

Getting Started User Guide 1.0

Date: October 24, 2004

Author: Rico Möckel

Supervisor: Prof. Auke Jan Ijspeert



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE



BIOLOGICALLY INSPIRED
ROBOTICS GROUP (BIRG)

Getting Started User Guide

Date	Version	Changes
October 24, 2004	1.0	-

Table of Contents

1 Preface	5
1.1 Purpose	5
1.2 Audience	5
1.3 Organization	5
1.4 Reference	5
1.5 Definitions	5
2 Acknowledgements	6
3 Idea of a modular robot	7
4 Constrains	8
4.1 Time	8
4.2 Costs and number of devices	8
4.3 Functionality / Flexibility	8
4.4 Usability	8
4.5 Energy	9
4.6 Place	9
4.7 Reusability / Technology	9
5 Hardware Overview	10
5.1 ZV4002	10
5.2 XC3S400-4TQ144C	10
5.3 Servo motor	11
5.4 Two Li-ION batteries	11
6 Organization of Electronics	12
6.1 Bluetooth board	12
6.2 FPGA board	13
6.3 Power board	14
7 Example: Controlling a YaMoR module with Bluetooth	15
7.1 Hardware requirements	16
7.2 Software requirements	16
7.3 Get the example running	17
7.3.1 Hardware setup	17
7.3.2 Building up a Bluetooth connection	22
7.3.3 FPGA programming	25
7.3.4 Controlling the YaMoR unit wireless	29
8 Next steps	32

List of Figures

Figure 1: A wheel build of six YaMoR units.....	7
Figure 2: A YaMoR module closed (left) and with opened case (right).....	10
Figure 3: Bluetooth board top and bottom.....	12
Figure 4: FPGA board top and bottom.....	13
Figure 5: Power board top and bottom.....	14
Figure 6: Data flow.....	15
Figure 7: Bluetooth board (bottom).....	17
Figure 8: Switch for powering YaMoR by battery.....	18
Figure 9: RS232 connector of Bluetooth board inside YaMoR.....	19
Figure 10: Switch for powering YaMoR by batteries.....	20
Figure 11: JTAG connector of FPGA board inside YaMoR.....	21
Figure 12: Create new connection in Hyper Terminal.....	22
Figure 13: Port settings.....	23
Figure 14: Invoke a command saved in a text file.....	24
Figure 15: ISE development environment from Xilinx.....	25
Figure 16: FPGA programming.....	26
Figure 17: Device selection.....	27
Figure 18: Start programming.....	27
Figure 19: Power supply for servo motor.....	28
Figure 20: Content of "traj.txt".....	29
Figure 21: Start the Serial Trajectory Transmitter.....	30
Figure 22: Serial Trajectory Transmitter.....	31

List of Tables

Table 1 References.....	5
Table 2 Definitions.....	5

1 Preface

This section explains the purpose of this document gives an overview about the audience and organization of the document and gives some information about references and definitions used.

1.1 Purpose

This document gives a first introduction to how to control a modular robot unit wireless by a PC. It is especially interesting for new users of YaMoR.

1.2 Audience

This document is written for new users of YaMoR the Modular Robot Unit from BIRG.

1.3 Organization

The sections 3 and 4 give a short overview about the modular robot project while section 5 and 6 explain some basic components and features of the robot module. Section 7 describes an example of how to use the different components inside the robot module together and shows the sequence of things to be done to control a robot module wireless by a PC. Section 8 gives a short outlook to help the user going on working with YaMoR.

1.4 Reference

In this document the following references are used.

Num	Document	Source
[1]	Pricing Spartan3	http://www.xilinx.com/prs_rls/silicon_spart/03142s3_pricing.htm
[2]	Bluetooth Board Reference Guide	BIRG
[3]	FPGA Board Reference Guide	BIRG
[4]	Power Board Reference Guide	BIRG
[5]	YaMoR document overview	BIRG

Table 1: References

1.5 Definitions

In the document the following definitions with the following meaning are used:

Definition	Meaning
BIRG	Biologically Inspired Robotics Group
YaMoR	Yet Another MOdular Robot

Table 2: Definitions

2 Acknowledgements

Even if only very few people were working directly on this project it would not have been successful if there would not have been a lot of friendly person supporting me.

First of all, I would like to thank Prof. Auke Jan Ijspeert for so much friendly advice, good tips and for making this internship possible for me. It was a lot of fun but also a great new challenge to develop this modular robot.

Doing an internship like mine in a foreign country with another language also means a lot of organization problems. That is why I would like to thank Madame Taric for all her help e.g. finding a nice home, her friendliness and all her organization work with regard to the modular robot project.

André Badertscher is a legend in terms of mechanics but also e.g. soldering electronics at the EPFL. He is not only responsible for many of the ideas for the mechanics of the robot module and for soldering all the electronics but he was also a good friend. Many thanks for all your help and pleasantness.

René Beuchat from the LAP was a great help in terms of designing the electronics, making the schematics and layouts. Thanks a lot.

Special thanks to Fabien Vannel for reviewing the schematics and for so much good advice in terms of electronics.

Peter Brühlmeier is working at the “Atelier de COncption, Réalisation et Test” (ACORT) at the EPFL. I am grateful that I could take advantage of his great experience in doing layouts and in terms of electronics.

Philippe Vosseler is the chef of the “Atelier de fabrication de circuits imprimés”. He gave me friendly advice for the layouts in terms of fabrication. For me it was very interesting to see the printed boards from this point of view and to familiarize with constrains for manufacturing e.g. boards for devices with ball grid arrays. Many thanks for your help.

I would like to thank Jonas Buchli for many funny but interesting conversations about the robot and for his good advice.

Alessandro Crespi is the expert for computer administration at BIRG but he is also an expert in many other things. I would like to thank him for many interesting conversations and for his help.

Thanks to Andres Upegui for his advice during the meetings. The first prototype of the modular robot unit was using his FPGA board.

I would also like to say thank you to all the other friendly persons at the LSL giving me a home here in Switzerland.

The first prototype of the modular robot is based on the BTnode from the BTnode Project. I would like to thank all the friendly people in Zurich supporting me with regard to the BTnode.

For me this project would not have been possible if there would not have been so many people at the Department of Electrical Engineering of the University of Rostock giving great lectures and preparing their students with a lot of knowledge and skills. Thanks a lot especially to Prof. Ralf Salomon, Matthias Handy, Jens Hildebrandt and Ronald Hecht for advice during the modular robot project.

Many, many thanks to my family and my girlfriend supporting me during the whole internship and all my studies before.

3 Idea of a modular robot

A modular robot is a robot consisting of a huge amount of units either identical or specialized for a certain problem. The idea behind YaMoR (Yet another Modular Robot) is to create a general purpose platform for e.g.

- Easily exploring locomotion,
- Developing new Bluetooth protocols,
- Doing research on FPGAs, using a MicroBlaze, doing partial reconfiguration and reconfiguring the FPGA via Bluetooth,
- Running distributed algorithm on controllers coupled wireless.

It should serve both as a platform for research on special technologies and algorithm e.g. for locomotion or for a research on a combination of different technologies e.g. a microcontroller and a FPGA but also as a tool easy to use. The developed elements could act as a tool where you just connect some robot modules with a servo and one degree of freedom to create a more complex modular robot. A user should be able to use the device without a lot of knowledge about wireless technologies just profiting of the fact that he does not have to handle a lot of disturbing cables. For those users a robot module will just be a toy or a simple tool with the necessary flexibility to meet different aims.

A simple example of how to connect the robot modules is shown in Figure 1. Six identical units are connected to build up a wheel.



Figure 1: A wheel build of six YaMoR units

4 Constrains

Designing embedded and small systems always means working with a lot of challenges. Some examples should be given here.

4.1 Time

Time was one of the key constrain for the project. Within six months we tried to develop a new modular robot unit including mechanical and electronic hardware, software and VHDL code. Furthermore a new scatternet protocol had to be designed.

During the project it was proven that even the best solution for a great device becomes completely impossible if e.g. the delivery time for a component is to long.

4.2 Costs and number of devices

The costs per device depend highly on the number of devices you want to produce. If you are only producing a prototype you can get a lot of electronic components as samples but the costs for the devices you have to buy increase. The best example is the Spartan3 inside the robot. If you buy 250000 devices Xilinx offers the XC3S400 for less than \$6.50 (2004 [1]). If you buy only 20 devices from a distributor a XC3S400 costs \$30.23. For a single device you even have to pay \$68.55.

On the other hand costs are not as critical if you build just one prototype. If you create more devices e.g. 10 all cost are automatically multiplied by 10. More prototypes normally also mean that it becomes impossible to get enough samples, because most companies only offer up to 5 samples. However, when designing YaMoR we tried to reduce cost as much as possible. E.G. to lower production cost we used only boards with two layers.

To keep the cost for the end user as small as possible we tried to design the robot also modular in terms of electronics. If you e.g. do not need an FPGA you do not have to take one. Just take the FPGA board out of the robot.

4.3 Functionality / Flexibility

Because the modular robot should be able to serve as a research platform it is designed to meet a lot of very different targets. That is why we decided to implement both a FPGA and a microcontroller for most possible flexibility. But this is also a reason why we decided to use a wireless technology that is providing a communication at every time to every module. A user should be able to control every element without connecting and disconnecting any wires that might even disturb the locomotion of a robot system. The modular system of electronics allows a user to replace just one of the three cards using the functionalities of the other boards as before.

4.4 Usability

The modular robot is also designed for users that potentially do not want to care about electronics and a lot of Bluetooth details but just run a software program for locomotion on a microcontroller or control the module via the FPGA and use some wireless and flexible connection. For those users the Bluetooth could be just a black box to which you send some data and get some data back in the same way like using a normal cable.

We would like to have a device where if you like you can just load some algorithm for locomotion, plug some modules together and see what happens. We could prove that with the tools and the modules we developed it is very easy to create some interesting locomotion. E.g. just write a little text file with angles for the servos and addresses of the modules you wish to use. Connect the modules mechanically and send the data with our tools to the robot. You can even use a simple graphical interface for selecting the modules and angles or create the trajectories for locomotion with MATLAB.

4.5 Energy

In embedded systems there is always the problem of limited energy. Huge energy consumption means always bigger systems with bigger batteries. These systems then need more energy just for moving their own load. But Energy consumption is also an important constrain for wireless communication. That is one reason why we were searching for a possibility to transmit data wireless without consuming a lot of energy.

4.6 Place

Within a robot module that should be as small as possible it is always a problem to realize all necessary functionality. Even devices that are considered to be small in industrial environment become too big. However, we are still working on making the robot module smaller.

4.7 Reusability / Technology

I tried to use technologies that should be supported for a long time like the Spartan3 and the ZV4002. Bluetooth seams to be a technology with good chances for later development. And the ZV4002 is already certified for the Bluetooth standard 1.2.

5 Hardware Overview

A YaMoR unit is shown in Figure 2.

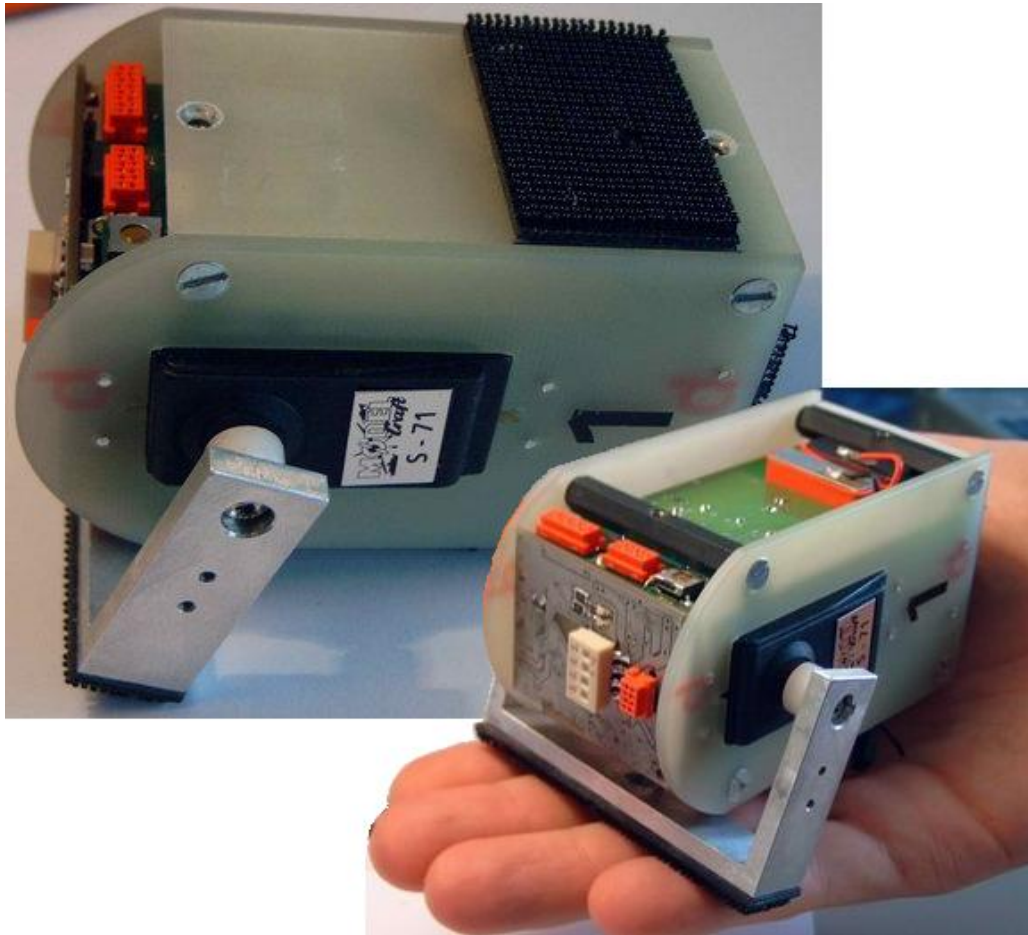


Figure 2: A YaMoR module closed (left) and with opened case (right)

Especially to meet the aim of functionality and flexibility the modular robot contains of the following core components:

5.1 ZV4002

This system on chip produced by Zeevo includes the integrated RF components for the Bluetooth and an ARM7TDMI which serves as an embedded processor.

Special features are that the ZV4002 supports scatternets with up to 7 slaves and up to 4 piconets. So a Bluetooth device can be both master of a piconet and slave of up to 4 other piconets. That is why the ZV4002 can be used to test new Bluetooth protocols where not only trees but also graphs with cycles are created.

5.2 XC3S400-4TQ144C

The modular robot includes a Spartan3 FPGA with 400000 gates meeting most requirements of digital logic for a lot of different applications. Loading a soft processor like a MicroBlaze and/or a PicoBlaze into the FPGA enables developers to have different microcontrollers inside the robot together with reconfigurable hardware.

5.3 Servo motor

For supporting locomotion with one degree of freedom every robot module includes a servo. The PWM-Signal for controlling the servo can be generated by the FPGA or the ZV4002.

5.4 Two Li-ION batteries

YaMoR contains two Li-ION batteries. Each battery typically provides 700mAh and a nominal voltage of 3.7V.

6 Organization of Electronics

The electronics inside the modular robot are distributed on three different boards:

- Bluetooth board
- FPGA board
- Power supply

Different advantages can be taken from this architecture.

1. Replacement or leaving out of not used electronics
If additional or other functionality has to be supported by hardware one of the boards can be replaced without developing the other boards a second time. So the user can take advantage of the already developed boards he wishes to use only changing what is really necessary. This lowers cost for electronic components and saves development time. Not used hardware can be left out of the robot. So the hardware is not using energy and the robot has less weight and load.
2. Easy reconfiguration of connections
GPIO pins of the boards can be connected and reconnected as needed without providing jumpers using a lot of space inside the robot.

6.1 Bluetooth board

An overview about the Bluetooth board is shown in Figure 3. The main component of this board is a ZV4002 an ARM Bluetooth combination. For more information about the Bluetooth board please see the Bluetooth Board Reference Guide [2].

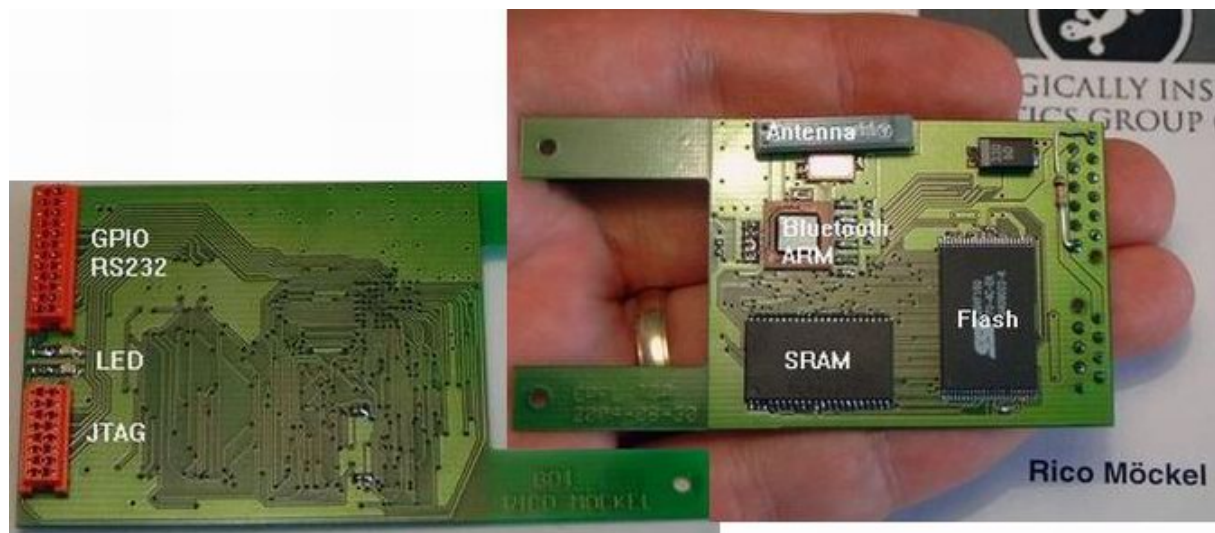


Figure 3: Bluetooth board top and bottom

6.2 FPGA board

An overview about the FPGA board is given in Figure 4. The main component of this board is a Spartan3 FPGA. For more information about the FPGA board please see the FPGA Board Reference Guide [3].

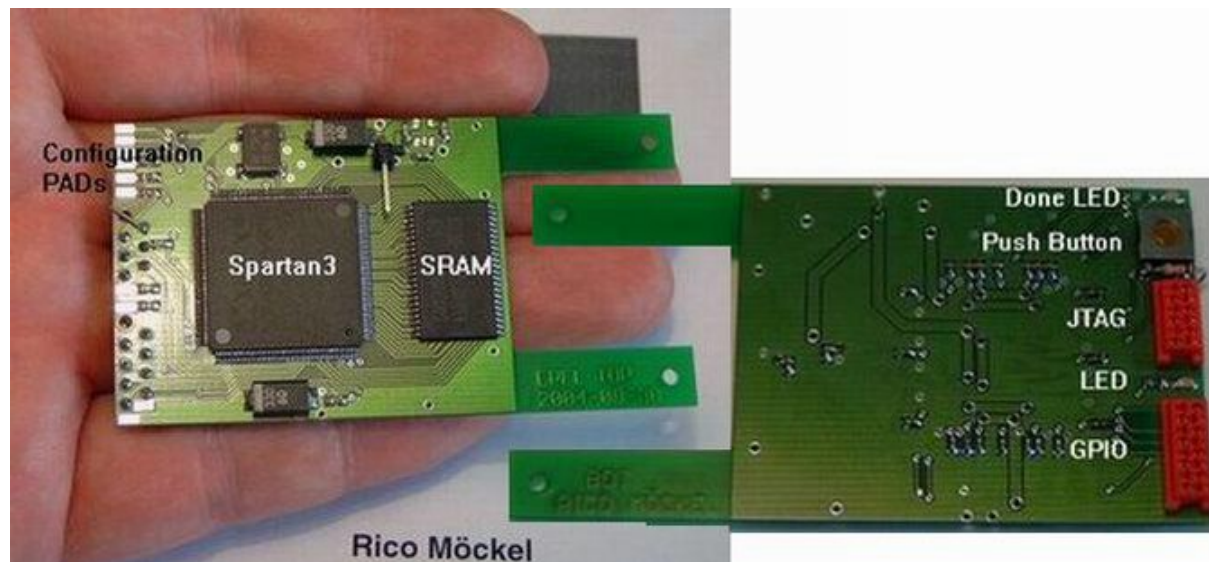


Figure 4: FPGA board top and bottom

6.3 Power board

An overview about the Power board is given in Figure 5. This board performs the power supply providing three different voltages and battery management like battery charging. For more information about the Power board please see the Power Board Reference Guide [4].

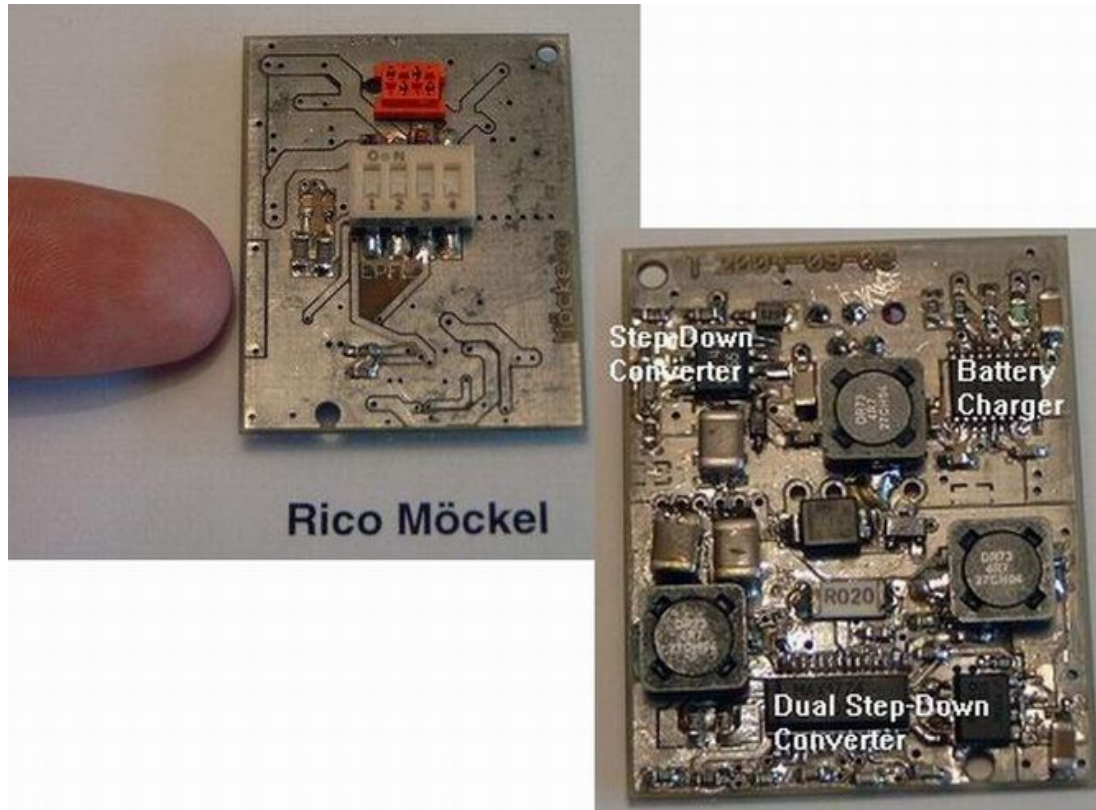


Figure 5: Power board top and bottom

7 Example: Controlling a YaMoR module with Bluetooth

The following description shows how to control a single YaMoR unit wireless by a PC using both the FPGA and the Bluetooth device inside the YaMoR unit. The data flow used in this example is shown in Figure 6.

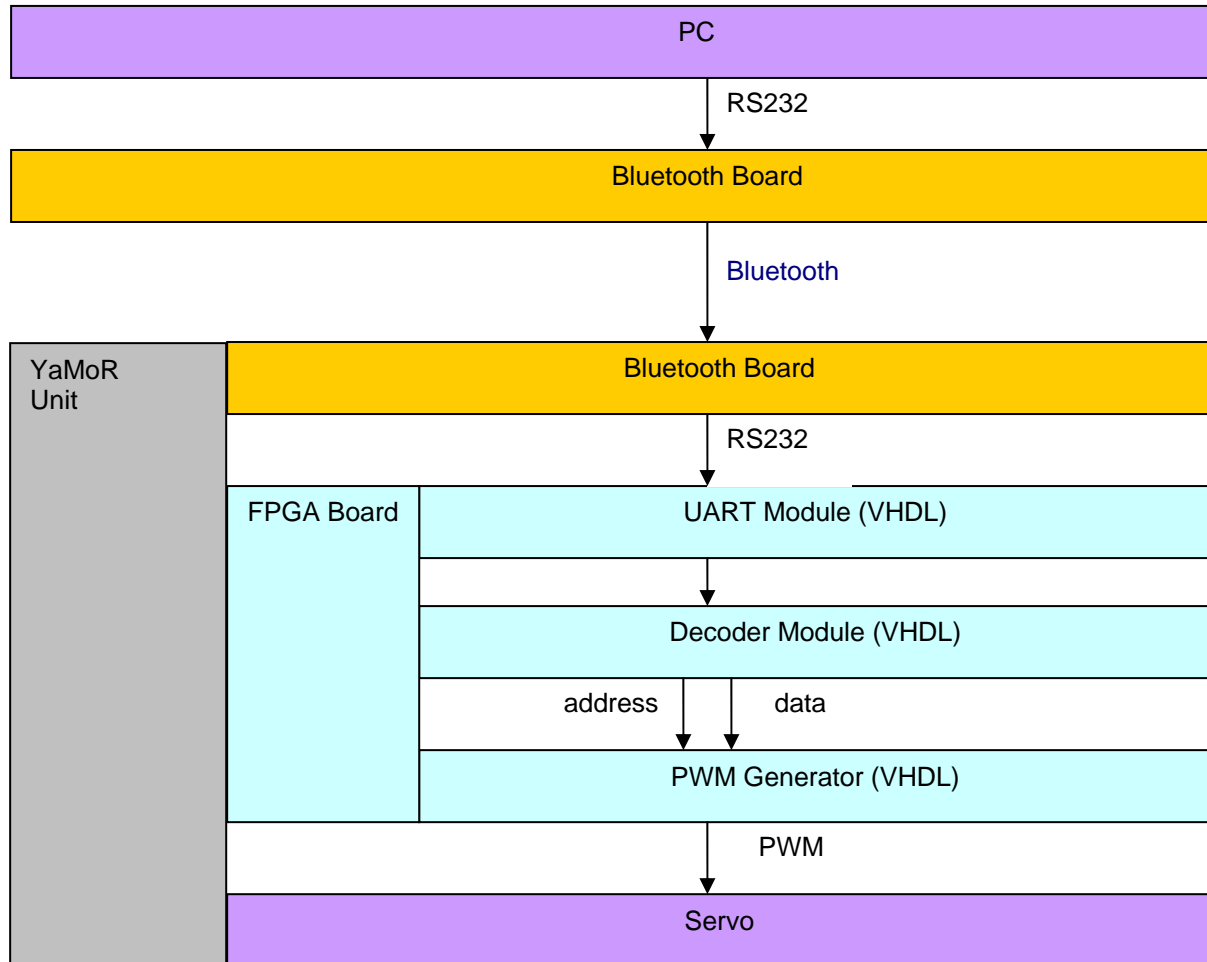


Figure 6: Data flow

The PC transmits the data for controlling the servo inside the YaMoR unit with help of a serial communication to a Bluetooth board. The Bluetooth board forwards the data wireless to the Bluetooth board inside the YaMoR unit. This data is sent via serial communication to the FPGA inside the YaMoR unit. Because the servo has to be controlled by a PWM signal the data received by the FPGA has to be transformed into a PWM signal. There for some VHDL program is loaded into the FPGA. This VHDL program makes it possible to receive the data via serial communication, interpret it and generate the PWM signal. So the servo inside the robot module can be directly controlled by the Spartan3.

7.1 Hardware requirements

For doing this example yourself you will need:

- A complete YaMoR unit
- A free Bluetooth board (Alternatively you can use a second YaMoR unit.)
- Power supply (3.3V) for the free Bluetooth board
- Serial communication cable to connect Bluetooth board and PC
- FPGA Programmer from Xilinx

7.2 Software requirements

For doing the example you need the following software:

- Serial Trajectory Transmitter (BIRG)
- Example trajectory (BIRG)
- VHDL program motor_uart_array_spartan3 (BIRG)
- Xilinx ISE Development Environment
- Microsoft Visual C++
- Hyper Terminal

All software with the remark “(BIRG)” is available at the project CD from Rico Möckel in the folder “Final_Presentation”.

7.3 Get the example running

The following sections show how to run the example.

7.3.1 Hardware setup

If you are using a free Bluetooth board:

- 1a. Connect the power supply (3.3V and ground) to the free Bluetooth board. **Please start with the ground.** The cable for the RS232 connector includes a connector for the power supply. Afterwards the green LED of the Bluetooth board should be blinking.
- 2a. Connect the free Bluetooth board to a serial port of a PC. The RS232 connector of the Bluetooth board is shown in Figure 7.

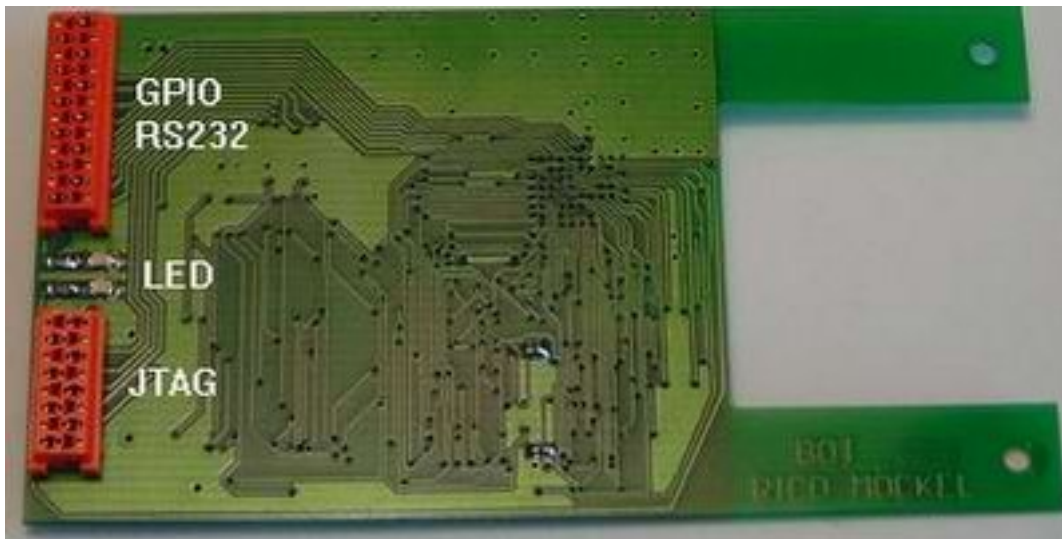


Figure 7: Bluetooth board (bottom)

If you are using two YaMoR units:

- 1b. Switch the first YaMoR unit on. The switch for powering the module with the help of the batteries is shown in Figure 8. Afterwards the green LED of the Bluetooth board should be blinking.

Please note: This switch is only for switching the electronics on not for the power supply of the servo. However, the power supply for the servo will not work if the electronics are not switched on, too.

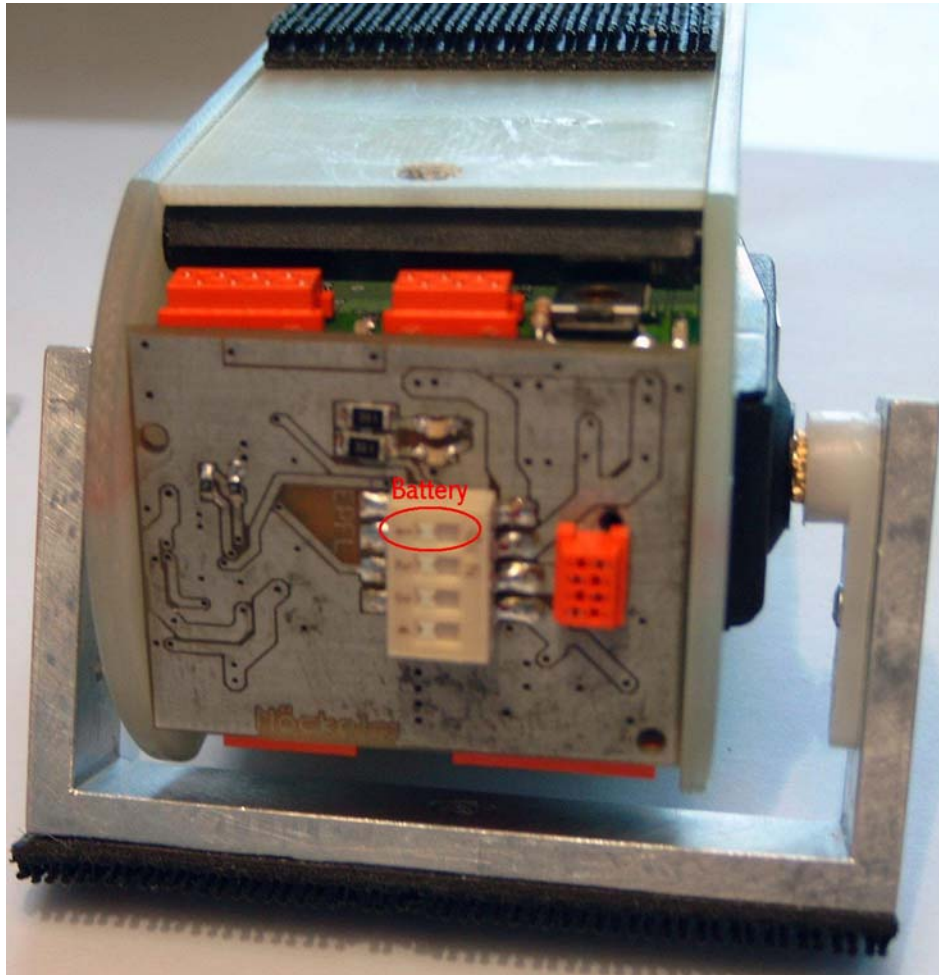


Figure 8: Switch for powering YaMoR by battery

- 2b. Connect the Bluetooth board inside the YaMoR unit to a serial port of a PC. The RS232 connector of the Bluetooth board is shown in Figure 9.

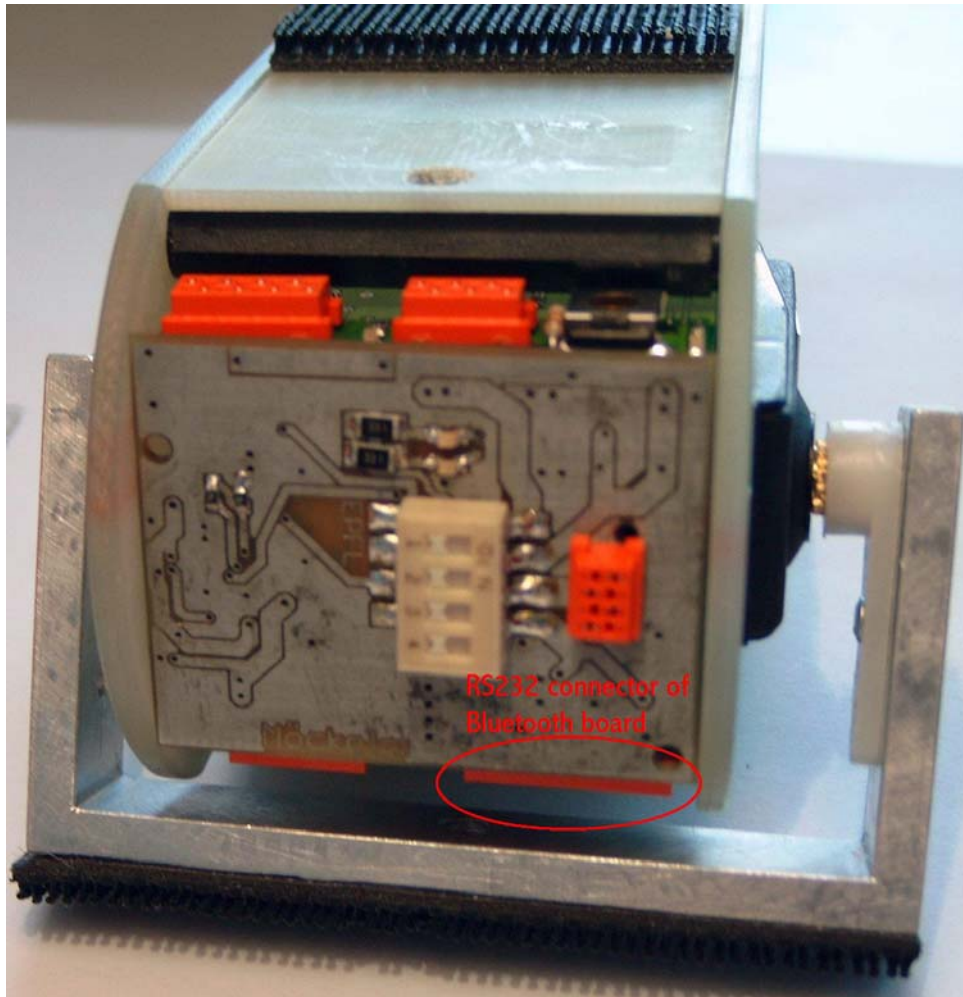


Figure 9: RS232 connector of Bluetooth board inside YaMoR

For both using a single YaMoR unit and a free Bluetooth board and for using two YaMoR units:

3. Switch the electronics of the YaMoR unit that should be controlled wireless on. The switch for powering the module with the help of the batteries is shown in Figure 10. Afterwards the green LED of the Bluetooth board should be blinking.

Please note: This switch is only for switching the electronics on not for the power supply of the servo. However, the power supply for the servo will not work if the electronics are not switched on, too.

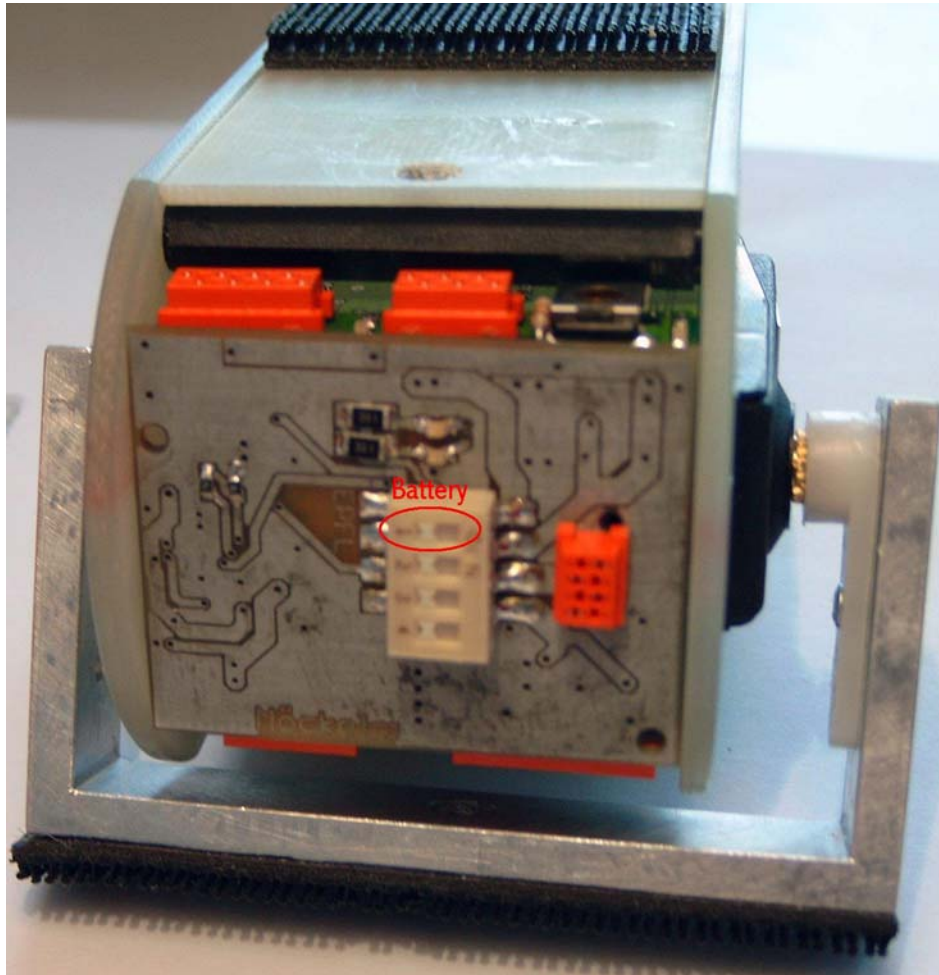


Figure 10: Switch for powering YaMoR by batteries

4. Connect the Xilinx programmer to the PC. Afterwards the LED of the programmer should be yellow.

5. Connect the JTAG connector of the FPGA board to the Xilinx programmer. The JTAG connector is shown in Figure 11. Please do not forget to connect also the 2.5V connector of the FPGA board to the reference voltage of the Xilinx programmer. Afterwards the LED of the programmer should be turned green.

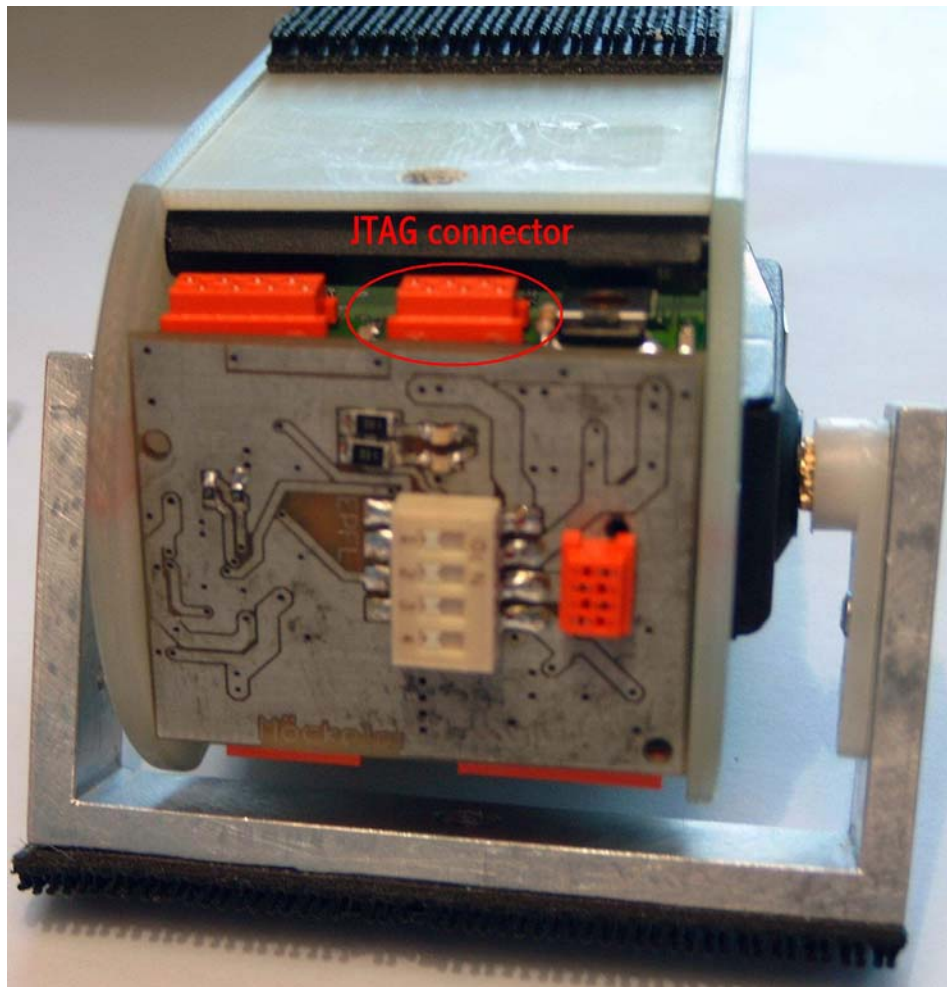


Figure 11: JTAG connector of FPGA board inside YaMoR

7.3.2 Building up a Bluetooth connection

When the hardware is connected the Bluetooth connection can be established. For this there must be established a serial connection either to the free Bluetooth board or to the YaMoR unit that is connected via RS232 to the PC. A short description of this procedure is given below.

1. Please start the “Hyper Terminal” and create a new connection as shown in Figure 12.

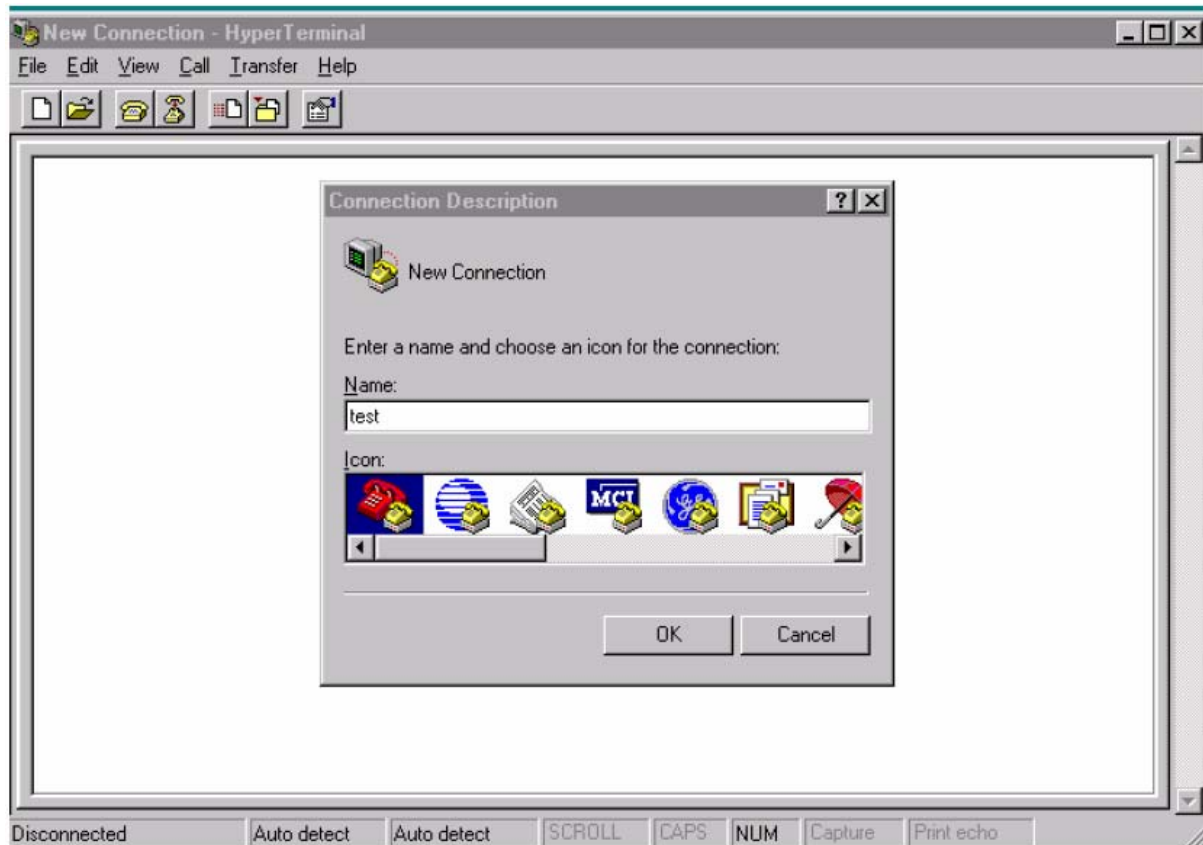


Figure 12: Create new connection in Hyper Terminal

- Please choose the port settings as shown in Figure 13. The baud rate should be selected to be 115200 bits per second.

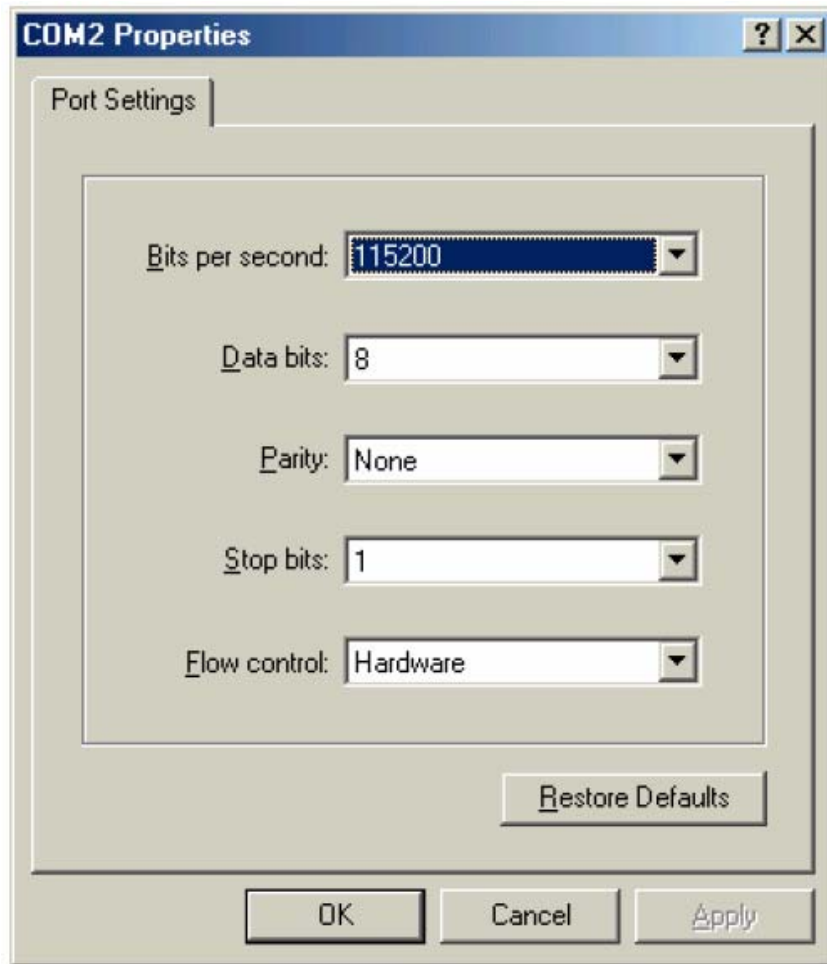


Figure 13: Port settings

When resetting the Bluetooth device (e.g. by switching the power supply off and on) it sends its Bluetooth address to the terminal and tells that it is in the command mode.

The Bluetooth device can be controlled by some text commands now. These commands can be typed directly into the terminal window. However, it is recommended to write them down into a text file and send them with the Hyper Terminal. You can find these text files in the folder "Final_Presentation" on the project CD of Rico Möckel.

With the command

AT+ZV Discovery

the Bluetooth device is asked to search for other Bluetooth devices. If this was successful it should print the Bluetooth addresses of the devices found to the terminal window.

Please note that there must be a new line/return/enter after the last word. This command can be found in the text file "discovery.txt".

- For invoking the discovery command with the help of a text file please click on "Transfer" in the Hyper Terminal main menu and select "Send Text File" as shown in Figure 14.

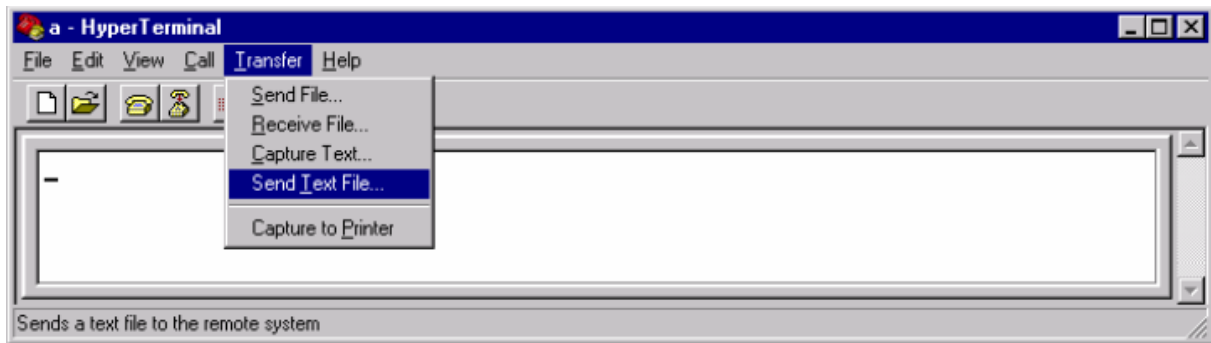


Figure 14: Invoke a command saved in a text file

4. Please select the text file “discovery.txt” and send it.

When the Bluetooth device inside the YaMoR module that should be controlled wireless is found establish a Bluetooth connection with the following command:

AT+ZV SPPCONNECT 00043e000000

where “00043e000000” should be replaced by the Bluetooth address of the device that should be connected. If this was successful the Bluetooth devices go into “Bypass Mode”. This means that everything typed into the Hyper Terminal will be directly send to the wireless connected device.

Please note that there must be a new line/return/enter after the last word. The connect command can be found in the text file “connect_command.txt”.

5. For invoking the connect command with the help of a text file please click on “Transfer” in the Hyper Terminal main menu and select “Send Text File” as shown in Figure 14.
6. Please select the text file “connect_command.txt” and send it.

When the connection is established the greed LED of the Bluetooth boards should stop blinking and stay switched on.

7. Please close the Hyper Terminal connection.

7.3.3 FPGA programming

After connecting the hardware and building up the Bluetooth connection the FPGA should be programmed.

1. Please start the ISE development environment from Xilinx by double clicking on the project file "motor_uart_array_spartan3.npl" that can be found in the folder "Final_Presentation/ motor_uart_array_spartan3" on the project CD of Rico Möckel.
2. Please double click on the field "Configure Device (IMPACT)" (please see Figure 15) to start the synthesis and implementation of the VHDL code.

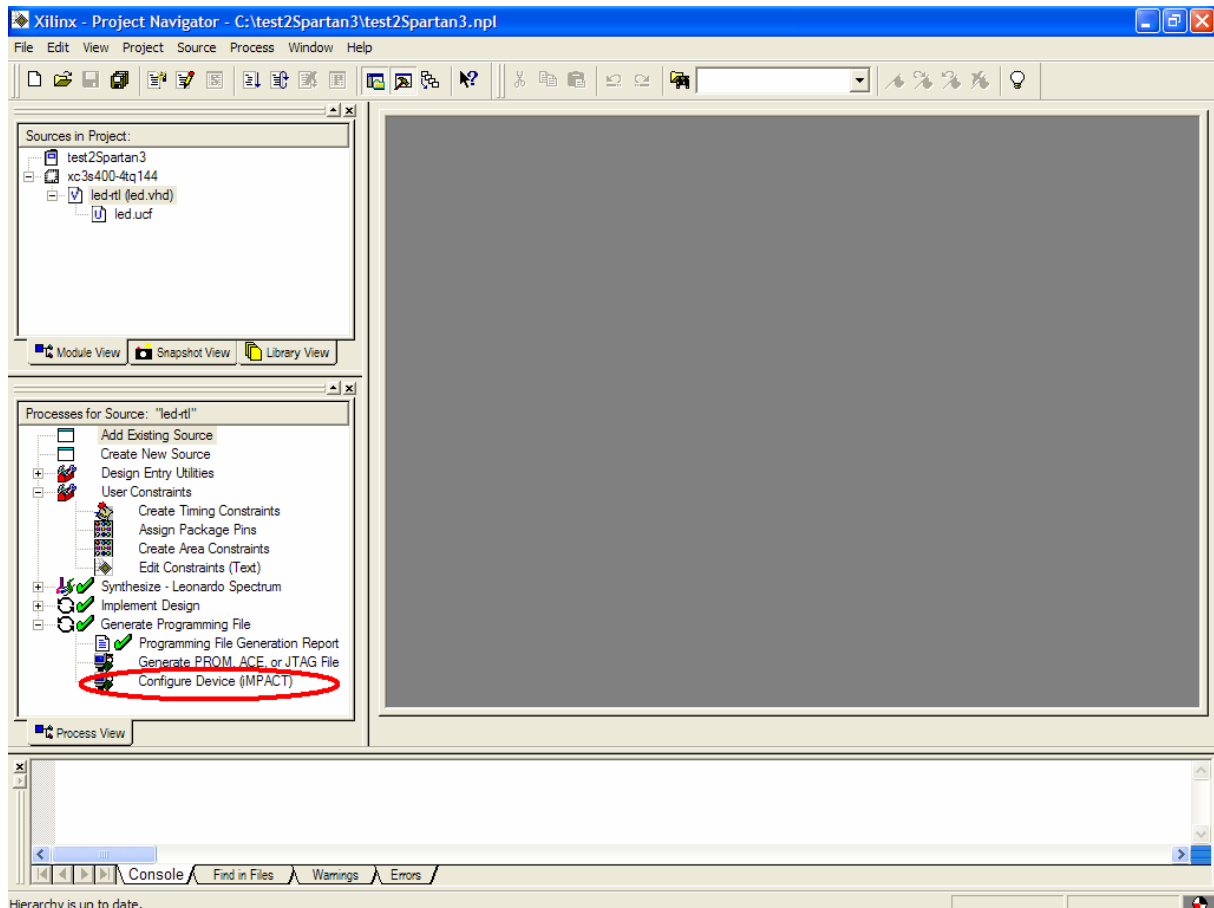


Figure 15: ISE development environment from Xilinx

3. Afterwards the FPGA can be programmed. Please select “Boundary-Scan Mode” as shown in Figure 16 and click next.

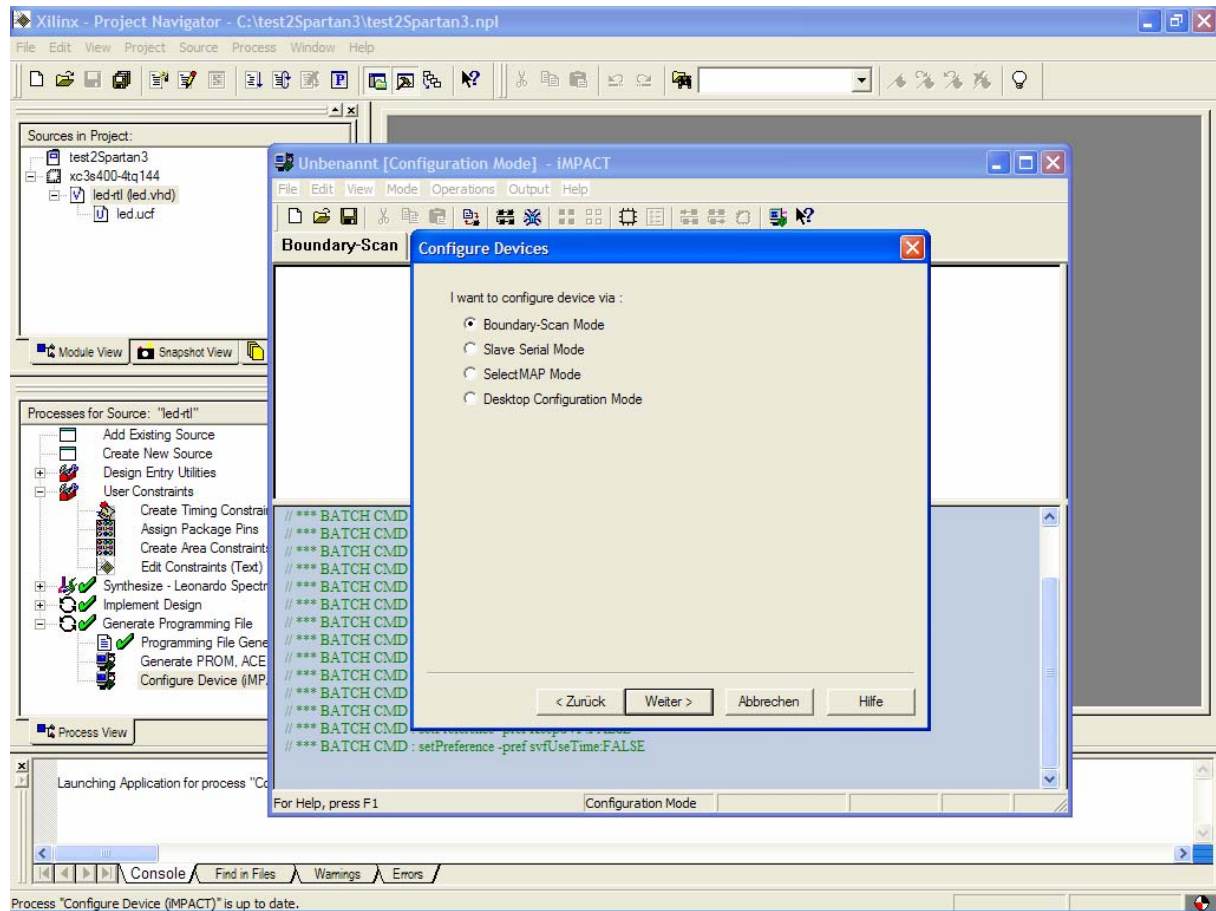


Figure 16: FPGA programming

The tool will search for a Spartan3 FPGA connected to the Xilinx programmer. It will tell you if the search was successful.

4. Please select the bit file “controller.bit” to be loaded into the FPGA.

- Please click with the right mouse button on the small circuit shown in Figure 17 and select "Program..." as shown in Figure 18.

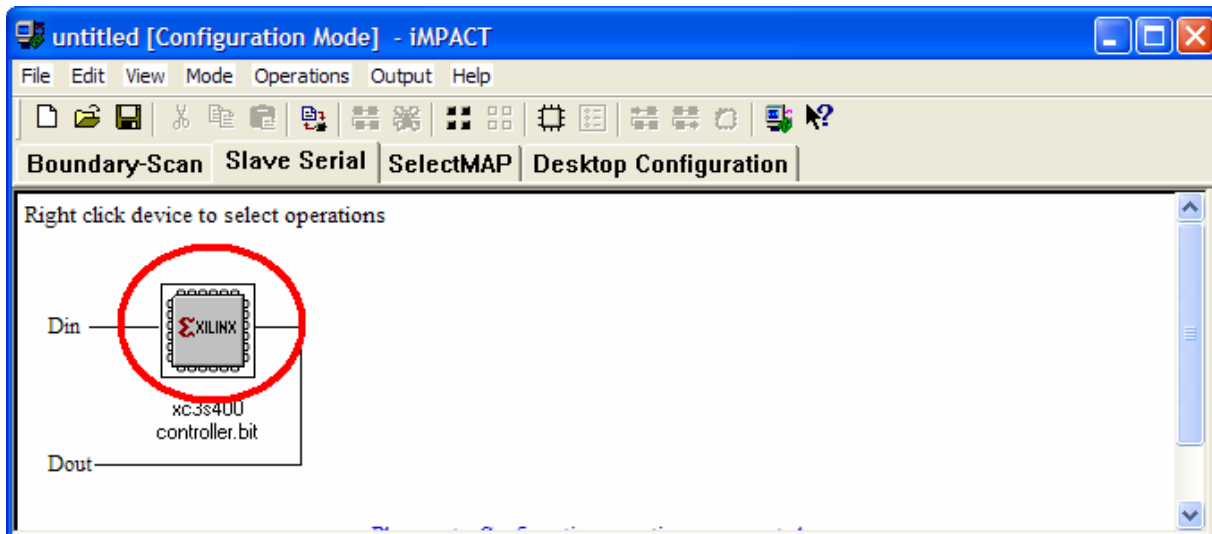


Figure 17: Device selection

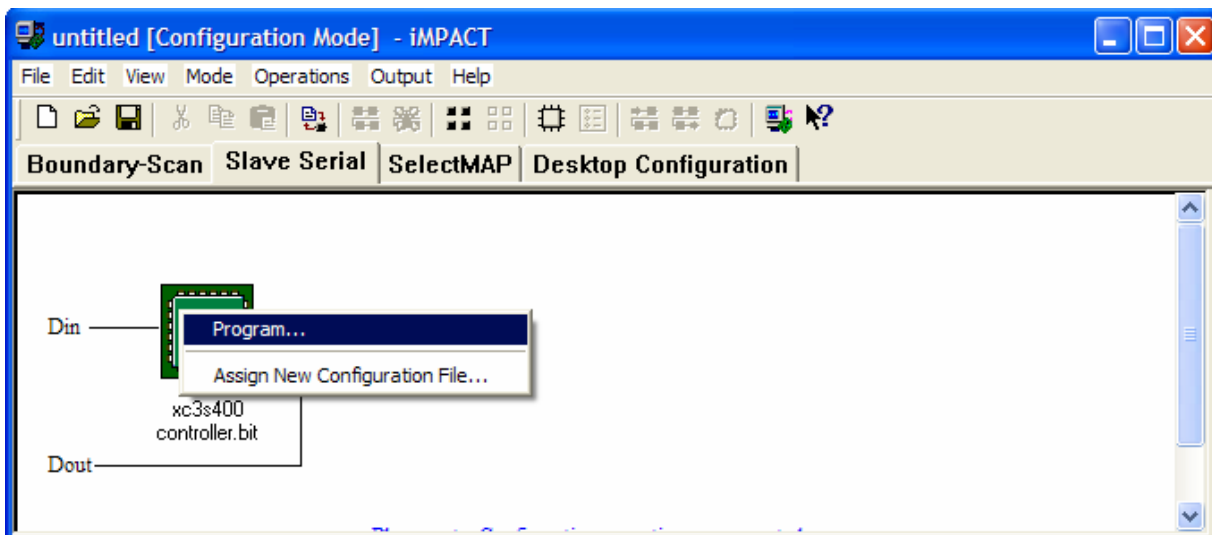


Figure 18: Start programming

- Please accept the following window. The programming starts.

If programming was successful the tool will give out a blue message "Programming succeed" and the green DONE-LED on the FPGA board inside the YaMoR module will turn on.

- You can disconnect the JTAG connector from the YaMoR unit now. Now the module is really wireless.
- Reset the FPGA by pressing the push button of the FPGA board.

9. Please turn the power supply for the servo on as shown in Figure 19.

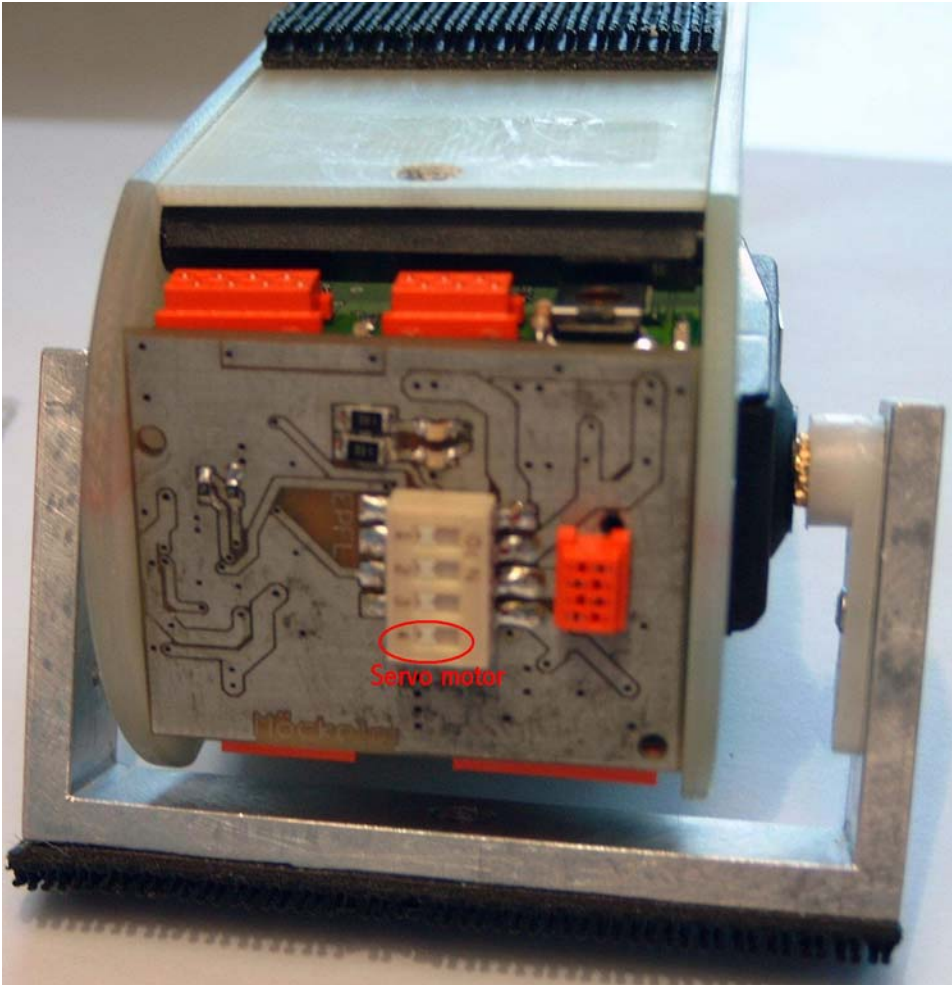


Figure 19: Power supply for servo motor

7.3.4 Controlling the YaMoR unit wireless

The last step is to send some trajectories to the YaMoR module.

Please note that the power supply for the servo has to be switched on before the module can move.

Please note that the Hyper Terminal connection has to be closed before starting the Serial Trajectory Transmitter.

If you are using “Microsoft Visual C++”

- 1a. Please double click on the project file “traj.dsw” to start the Microsoft Visual C++. The project file can be found in the folder “Final_Presentation/trajectory-rs232” on the project CD of Rico Möckel.

The trajectory is saved in the text file “traj.txt” that also could be found in the same folder. The content of the file is shown in Figure 20.

```
<receiver> 255 </receiver>
<set_delay> 500
<set_time_factor> 1.0
<loop> 1 100
10
0
-85
</loop> 1 1
```

Figure 20: Content of "traj.txt"

It tells the tool to send every 500ms a new value to the Bluetooth device. The angles of the Aluminum-U of the YaMoR module are directly written down into the text file in a loop. The Aluminum-U will start with an angle of 10 degree. It will go back to 0 degree and afterwards to -85 degree. This loop is performed 100 times.

2a. Please start the Serial Trajectory Transmitter by pressing the exclamation mark shown in Figure 21 in the Microsoft Visual Studio C++.

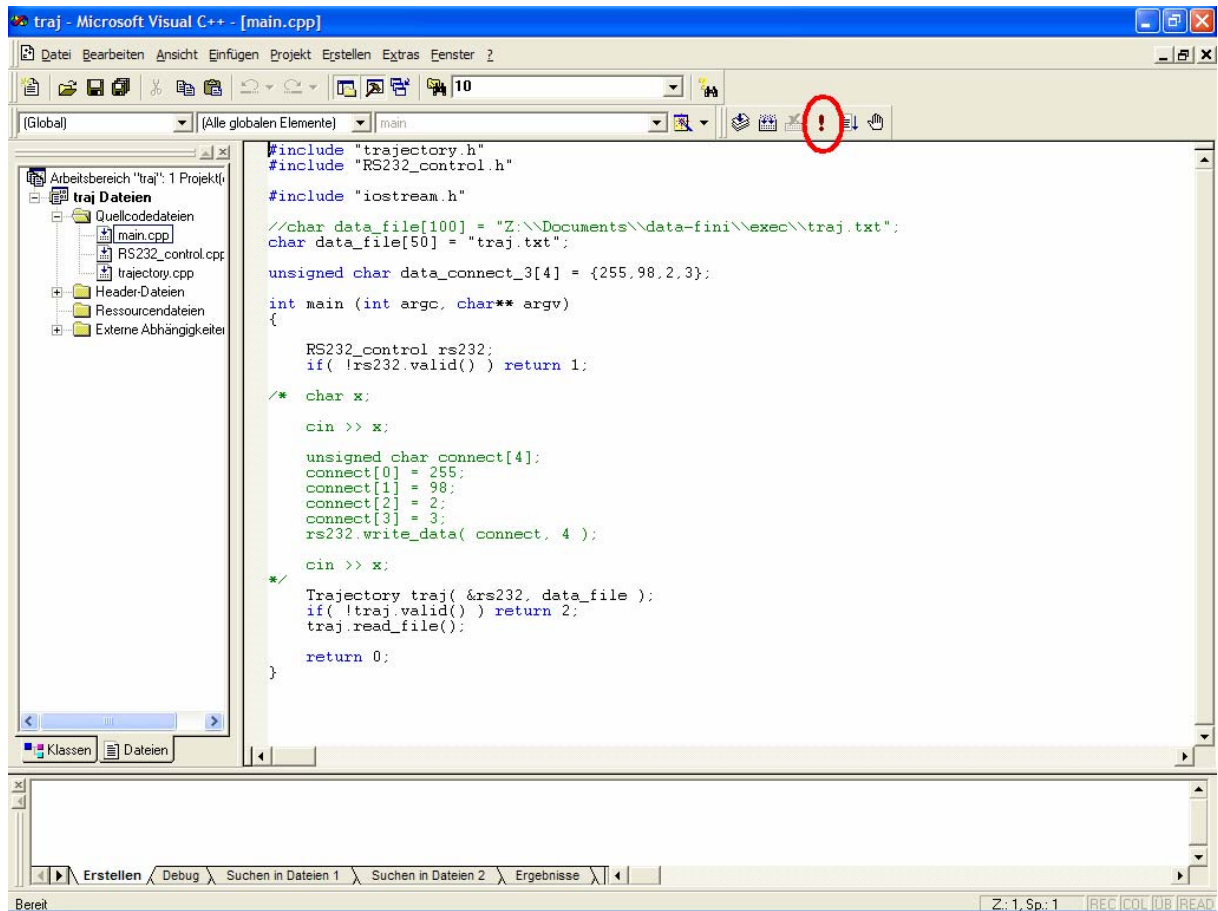
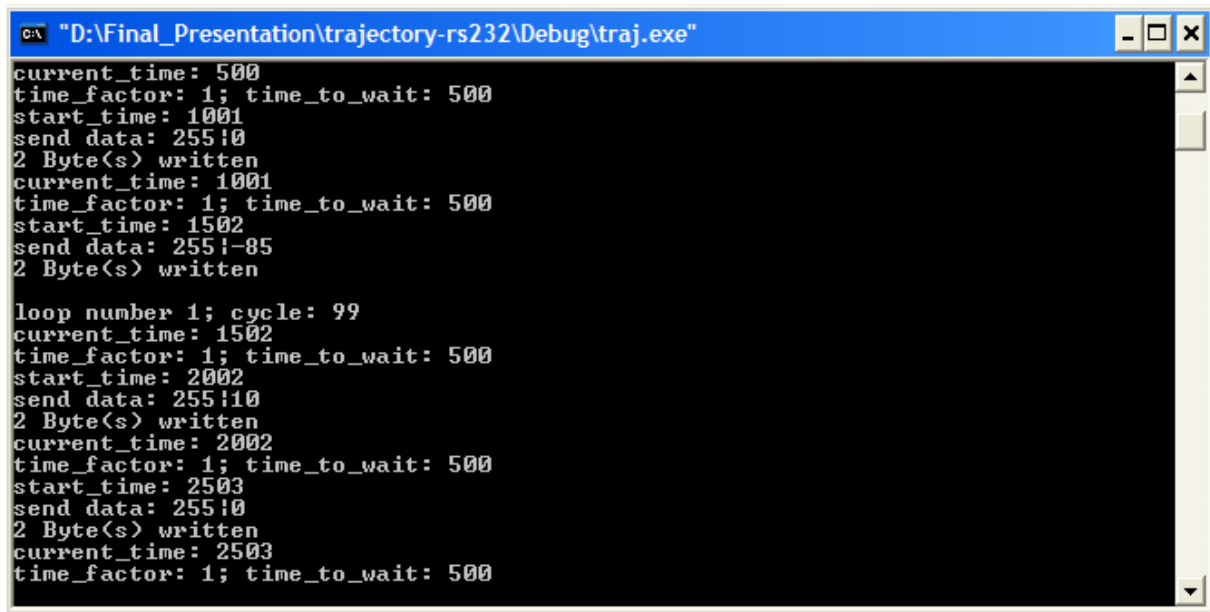


Figure 21: Start the Serial Trajectory Transmitter

A window shown in Figure 22 will appear that shows the values transmitted via RS232.



```
"D:\Final_Presentation\trajectory-rs232\Debug\traj.exe"
current_time: 500
time_factor: 1; time_to_wait: 500
start_time: 1001
send data: 255!0
2 Byte(s) written
current_time: 1001
time_factor: 1; time_to_wait: 500
start_time: 1502
send data: 255!-85
2 Byte(s) written

loop number 1; cycle: 99
current_time: 1502
time_factor: 1; time_to_wait: 500
start_time: 2002
send data: 255!10
2 Byte(s) written
current_time: 2002
time_factor: 1; time_to_wait: 500
start_time: 2503
send data: 255!0
2 Byte(s) written
current_time: 2503
time_factor: 1; time_to_wait: 500
```

Figure 22: Serial Trajectory Transmitter

The Aluminum-U of the YaMoR unit should move controlled by the PC.

If you are not using “Microsoft Visual C++”

- b. You can also start the executable file of the Serial Trajectory Transmitter directly by double clicking on the file “traj.exe”. This file can be found in the folder “Final_Presentation/trajectory-rs232/Debug”.

Note: Please verify that the right trajectory file “traj.txt” is in the same folder.

8 Next steps

When doing successfully the example before you may wish to play a little bit with the module by changing the values of the angles or time delay written down in the file "traj.txt".

For more information about available documentation for YaMoR please see the YaMoR document overview [5].